英文文本机器简化算法

An Algorithm for Machine Simplification of English Text

曹起曈　乔明达

南京外国语学校

Qitong CAO, Mingda QIAO

Nanjing Foreign Language School

江苏省，中华人民共和国

Jiangsu, P. R. China

2013. 8.

目录

摘要/Abstract **3**

1 引言 **4**

2 机器翻译面临的问题及其发展历程 **4**

2.1 机器翻译面临的问题 4

2.2 基于语言学的机器翻译算法及其评估 5

2.3 基于统计学的机器翻译算法及其评估 9

2.4 小结 10

3 基于词汇简化的“基本英语”的适应性 **11**

4 “基本英语”词汇数目的确定 **12**

4.1 简化方案的假设与单词的计数 12

4.2 语料库的选择 12

4.3 对词频与次序关系的分析 12

4.4 对“基本英语”词汇数目与文本覆盖率关系的分析 15

4.5 “基本英语”词汇数目的确定 16

4.5 其他参考标准 18

4.6 小结 18

5 原始文本中非“基本英语”词汇的替换方法 **18**

5.1 确定“基本英语”对原始文本词汇的表述 18

5.1.1 简化（释义）的原则 18

5.1.2 简化（释义）的方法和范例 19

5.1.3 维基式公测 20

5.2 确定语义合理性的统计学模型及其评估 20

5.2.1 n-gram 语言模型 20

5.2.2 n-gram 模型的问题 21

5.2.3 n-gram 模型的改进 21

5.3 n-gram 语言模型的应用 23

5.4 例外及其解决办法 25

5.4.1 屈折词缀 25

5.4.2 专有名词和术语 25

5.4.3 其它较为专业的名词 25

5.4.4 成语动词 25

5.4.5 对简化（解释）可用词汇的附加说明 26

5.5 小结 26

6 算法（伪代码） 26

6.1 函数说明 26

6.2 “基本英语”词汇判断算法 27

6.3 n-gram 语言模型算法 27

6.4 词汇简化算法 28

6.5 小结 28

7 参考资料 29

附录 30

作者介绍 36

摘要

本文评估了现有的机器翻译算法，并针对其存在的不足提出了“英文文本机器简化”方案. 我们通过对美国当代英语语料库（COCA）词频数据的分析得出了“基本英语”词汇数目与文本覆盖率之间的关系，并以此为基础确定了“基本英语”词汇的范围. 我们综合运用语言学和数学知识，明确了用“基本英语”简化正常英文文本的原则，利用完善后的 n-gram 模型解决语义识别问题，并给出了初步的算法. 此外，本文还以众多例证对机器翻译涉及的问题进行了简明的介绍.

关键词

计算语言学　机器翻译　“基本英语”　n-gram 模型　语料库　语义识别（词汇差异处理）

Abstract

This paper evaluates several existing methods of Machine Translation (MT), and, having addressed their shortcomings, proposes the idea of Machine Simplification of English Text. We established the relationship between the number of words in “Fundamental English” and its coverage of normal English text based on our analysis of word frequency data from the Corpus of Contemporary American English (COCA), and thus determined the range of “Fundamental English” vocabulary. Applying linguistic and mathematical knowledge, we laid down the principles of simplifying normal English words with “Fundamental English,” utilized a refined version of n-gram model to solve problems concerning semantic identification (word sense disambiguation), and provided a basic algorithm. In addition, this paper includes a concise and richly exemplified introduction to various issues relating to MT.

KEY WORDS

computational linguistics, machine translation (MT), “Fundamental English”, n-gram model, corpus, semantic identification (word sense disambiguation)

1 引言

互联网技术的发展使得信息的广泛传播成为可能,但目前全世界绝大多数网页均以英文写成，一定程度上阻碍了非英语国家互联网用户获取多元资讯. 机器翻译可以以低成本、高效率的方式减少语言壁垒，促进信息在全球范围的自由流通. 然而，目前基于统计的机器翻译算法尚不能很好地处理部分原始语言和目标语言间的形态句法差异,致使翻译结果不流畅甚至不正确，无法令用户满意. 另一方面，英语教育已在全球普及，对于非英语国家的互联网使用者（主要是接受过中学教育的青年），英语句法并不复杂. 但英语词汇数量庞大，艰深的词汇是非英语国家互联网使用者理解英文文本的主要障碍. 而目前的机器翻译算法可以较好地实现以词为单位的跨语言转化. 综合考虑这些因素，我们提出了一种全新的机器翻译模式——“英文文本机器简化”，即将正常的英文文本（源语言）中较为困难的单词替换成语义相同的“基本英语”单词，并输出简化后的文本（目标语言）.

2 机器翻译面临的问题及其发展历程

2.1 机器翻译面临的问题

机器翻译，指利用计算机软件将一种自然语言声音或文本转换为另一种自然语言. 本文中的机器翻译特指文本处理. 从理论语言学的角度看，书面语言间的差异主要由形态、句法和词汇语义三方面的差别导致. 各语言词汇形态复杂程度不同，有的语言词汇中包含丰富的词缀作为语素（综合性强），也有的语言不包含这样的词缀（综合性弱），但计算机程序可以将词缀、词根和（独立的）单词视作相同的词汇单位，所以形态综合程度的差别对机器翻译的影响较小. 因此，机器翻译主要面临句法（语序）和语义（词汇）上的问题.

语序差异是指源语言和目标语言在词汇单位顺序上的不同. 机器翻译时，程序不仅要在内建词典中寻找所涉及的词汇单位，同时还要决定输出的目标语言中这些词汇单位的顺序.

词汇差异是指有时源语言中某一词汇单位可能具有多种语义，而这些语义在目标语言中由不同词汇单位表示，因而源语言到目标语言的映射无法形成单值对应. 机器翻译时，程序必须判断目标语言中何种义项可以与源语言的词汇单位匹配.

2.2 基于语言学的机器翻译算法及其评估

科学家对机器翻译的有效尝试始于 20 世纪中期. 1954 年，乔治城大学研究者和 IBM 公司联合主导了俄-英机器翻译实验（Georgetown-IBM 实验），测试了 60 多个俄语语句，大部分与化学工业有关，但也包含了一些大众化的内容. 例 1 是 Georgetown-IBM 实验中输入的一个俄语语句、其每个单词的语义和 IBM 701 计算机给出的机器翻译结果：

例 1

Vyelyichyina	ugla	opryedyelyayetsya	otnoshyenyiyem	dlyini	dugyi	k	radyiusu.
size/value	of angle	is determined	by relation	of length	of arc	to	radius

Magnitude of angle is determined by the relation of length of arc to radius.

‘角的大小由其弧长与半径的关系决定. ’

下面由例 1 说明 Georgetown-IBM 实验实现句法差异处理和词汇差异处理的算法. 例 1 中的俄语单词 ugla 包含 2 个词汇单位，词根 ugl（‘角’，在内建词典中译为 angle）和属格后缀-a（在内建词典中译为 of），语序不同，体现了句法差异；例 1 中俄语词根 ugl 既可以表示‘角’（在内建词典中译为 angle），也可以表示‘煤炭’（在内建词典中译为 coal），体现了词汇差异.

为了解决这两个问题，Georgetown-IBM 实验的研究人员建立了含有 250 个俄语词汇单位的词典，每个俄语词汇单位最多附有 3 个编码，并可以对应 1 个或 2 个英语翻译. 研究者同时设计了 6 条语法规则，让程序通过检测每个词汇单位的编码判断语序和义项. 程序处理例句 1 中 ugla 一词时涉及的词条与算法可以简化表示如下：

内建词典（部分）

RU	EN1	EN2	CODE1	CODE2	CODE3
-a	of		131	222	25
ugl-	coal	angle	121		25

语法规则（部分）

```
#RULE 2 （目前 i=2，表示目前的词汇单位 ugl 是句中的第 2 个词汇单位）
if CODE1(i)=121 then
    if CODE2(i+1)=221 OR 222 then
        if CODE2(i+1)=221 then OUTPUT(i) ← EN1(i)
                               i ← i+1
        else OUTPUT(i)=EN2(i)
             i ← i+1

#RULE 3（目前 i=3，表示执行 RULE 2 后到达句中第 3 个词汇单位-a）
if CODE1(i)=131 then
    if CODE3(i-1)=23 then OUTPUT(i)← EN2(i)
                          i ← i+1
    else OUTPUT(i) ← EN2(i)
         swap OUTPUT(i), OUTPUT(i-1)
         i ← i+1
```

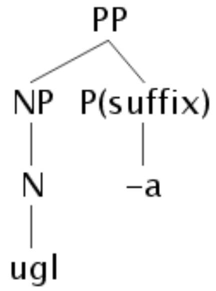
从算法中可以看出，Georgetown-IBM 实验中鉴别多义词义项的方法纯粹基于形态学：俄语中，‘煤炭’和‘角’两词的属格形式分别为 uglya 和 ugla，这使得程序可以通过属格后缀（为-ya 还是-a）判断出词根 ugl 的语义. 然而，这种做法并没有解决实质问题，因为大多数一词多义的例子都不具备形态句法识别特征[1]. 此外，该算法限定了每个源语言词汇单位至多只能对应 2 个目标语言义项，在实际处理中远远不够[2]. 所以，该算法在处理词汇差异的一般问题上无效.

该算法在处理句法差异时的方法虽然也基于语言学，但相较而言更为可取. 在俄语和英语中，分别有如下的句法规则：

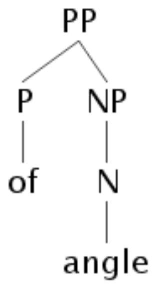
俄语：PP→NP P(case suffix) （介词短语→名词短语 介词(格后缀)）

[1] 例如，英语中 pen 作为名词，可以表示‘家畜的圈’或‘笔’，两者在形态句法上没有差别.

[2] 例如，pen 作为名词，《韦氏词典》（Merriam-Webster’s Collegiate Dictionary, 11th edition）中就给出了 8 个义项.



英语：PP→P NP （介词短语→介词 名词短语）



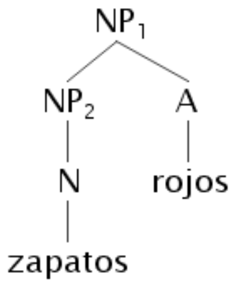
因此，在从俄语翻译成英语的过程中，俄语名词的属格后缀翻译成英语介词 of 时须置于名词前是一项普遍规律.

但是从语言学角度解决句法差异也并非轻而易举. 首先，各语言的语序存在明显差异，且任两对语言差异的方面都不同，因此机器翻译程序的算法不具有普遍性. （如果选择一种语言 C——例如英语——作为源语言 A 和目标语言 B 之间的过渡语言，算法可以得到相当程度的简化. [3]但不可避免地，两次机器翻译的过程会使得翻译质量显著下降. [4]）其次，由于结构歧义的存在，即使对于语言学家十分了解的两种语言，机器语序处理也可能产生差错. 例如，西班牙语和英语对于名词短语分别有如下的生成规则：

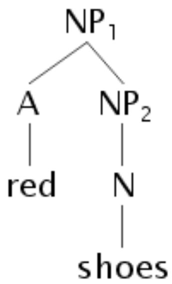
西班牙语：NP→NP A （名词短语→名词短语 形容词）

[3] 目前 Google 翻译虽然基于统计学方法，但其跨语言处理的过程也涉及英语作为过渡语言. 例如，从法语翻译成汉语时，算法会要求程序先将法语翻译成英语，再将英语翻译成汉语.

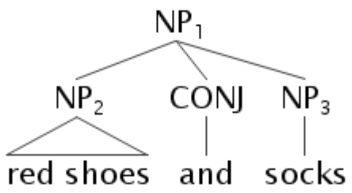
[4] 例如，Google 翻译将法语的 vous（'您'）翻译成汉语'你'. 法语和汉语对于第二人称单数皆有尊敬式（vous、您）和一般式（tu、你）的差别（T-V distinction），但英语缺乏这种区别（两者皆为 you）. 因此，Google 翻译算法先将'vous'翻译成'you'，再将'you'翻译成'你'，从而造成翻译错误.

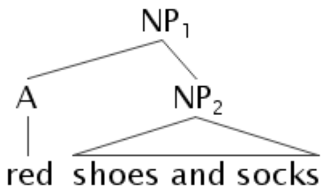


英语：NP→A NP （名词短语→形容词 名词短语）



基于此，英-西机器翻译算法在处理语序时理应设计将名词短语和形容词顺序调换. 但这样并不能保证语义准确传达. 例如，‘red shoes and socks’这个英语短语可能有 2 种不同的句法结构，其语义也不相同[5]：





因此，将‘red shoes and socks’翻译成西班牙语时，为保留结构歧义可能产生的所有语义，理想情况下须按照例 2 翻译：

例 2

calcetines y zapatos rojos

socks and shoes red

red shoes and socks

‘红色的鞋子和袜子’

换言之，此处不仅涉及名词短语和形容词语序的替换，还涉及并列连词连接的名词短语的语序替换. 例 2 的处理过程说明，通过语言学方法处理句法差异，理论上是可能的，但实

[5] 在袜子是否是红色这一问题上产生歧义.

际操作复杂. 考虑到处理算法只针对特定一组语言有效，没有普遍性，这样的处理在我们的“文本简化”方案中并不划算.

2.3 基于统计学的机器翻译算法及其评估

统计机器翻译的概念早在 1949 年即被提出，但由于 Georgetown-IBM 实验基于语言学规则算法的“巨大成功”和随后乔姆斯基在《句法结构》一书中对量化分析方法的驳论，统计算法长期遭受轻视. 20 世纪 80 年代，很多研究者充分意识到了基于语言学规则的机器翻译算法不能有效解决词汇语义处理问题，由此开始另辟蹊径，运用和改进统计学算法，在语义识别和句法识别上都取得了一定成就，但也存在许多突出的问题.

统计机器翻译基于大量的双语平行语料（源语言和目标语言之间的互译语句）. 通常情况下，研究人员首先按照语料内容的领域（domain）、主题（topic）和模式（modality）为平行语料分类[6]. 之后，研究人员将平行语料进行语句排列（sentence alignment），进而统计出源语言语句 $a=(a_1, a_2, ..., a_{l_a})$ 中的一个词汇单位 a_i 在目标语言中所有可能的对应词 $b_{i_1}, b_{i_2}, ..., b_{i_n}$ 出现的概率 $p(b_{i_1}|a_i), p(b_{i_2}|a_i), ..., p(b_{i_n}|a_i)$. 接着，选取所需翻译内容的领域、主题和模式下，目标语言中概率最大的对应词，以实现词汇差异的处理.

句法差异的处理则更为复杂. 纯粹的统计算法通常涉及一个排列函数，以处理源语言 a 和目标语言 b 之间词序的不同. 我们用 $i \to \alpha(i)$ 表示目标语言语句 b 中第 i 个词对应源语言语句 a 中的第 $\alpha(i)$ 个词. 运用处理词汇差异时的概率方法，可以得到，当语句 a、b 长度一定时，将 a 翻译成 b 的概率

$$p(b,\alpha|a) = C\prod_{i=1}^{l_b} p(b_i|a_{\alpha(i)}),$$

其中 C 为与 a,b 长度 l_a, l_b 有关的定值. 显然，随着 l_a, l_b 的增加，$p(b,\alpha|a)$ 减小. 最初

[6] 这样可以确保青少年的日常对话、科学文献和联合国决议在翻译时得以分别处理.

的 IBM 统计机器翻译算法即是基于下式判断 $p(b,\alpha|a)$ 的：

$$p(b,\alpha|a)=\frac{\varepsilon}{(l_a+1)^{l_b}}\prod_{i=1}^{l_b}p(b_i|a_{\alpha(i)})$$

算法输出 $p(b,\alpha|a)$ 最大的 $b=(b_1,b_2,...,b_{l_b})$ 作为翻译结果.

此种算法与 Georgetown-IBM 实验中的算法相比，在词汇差异处理方面有很大的进步. 但对于特定文本而言，虽然其句法规则可能与其上下文无关（因此 2.2 节中涉及的上下文无关语法算法理论上可处理句法差异），但其中某个单词的语义往往与其上下文（尤其是其前后的词语）有关. 这种统计算法只考虑了源语言文本内容领域、主题和模式对词汇语义的影响，而没有考虑具体的上下文对词汇语义的影响，因而需要改进.

然而，在句法处理的部分，这种基于统计学的方法不能实现基于语言学方法所达到的准确性. 首先，算法忽视语法规则带来的部分限制条件，因此当 $\prod_{i=1}^{l_b}p(b_i|a_{\alpha(i)})$ 达到最大时，不能保证语句 b 中的单词顺序符合句法要求. 其次，就该模型的 $\frac{\varepsilon}{(l_a+1)^{l_b}}$ 而言，当语句长度很大时，虽仍可比较不同的 p 的相对大小而输出最大的 $p(b,\alpha|a)$ 对应的 b，但由于最大的 $p(b,\alpha|a)$ 也很小，最终输出的结果准确性很难保证.

2.4 小结

机器翻译主要面临句法差异处理和词汇（语义）差异处理这 2 个问题，前者理论上可依靠语言学中的生成句法规则解决，但较为复杂，对于我们的“文本简化”成本太高；后者可在一定程度上依靠统计学方法解决.

3 基于词汇简化的“基本英语”的适应性

考虑到目前尚无法在任两种语言间轻松实现句法差异处理，我们提出了一种过渡性的方法：将复杂的英文文本转化成简易的“基本英语”文本.“基本英语”词汇由英语中最常见的 n 个单词组成. 算法可以将正常英语文本中超出“基本英语”词汇的单词通过统计算法替换成语义相同的“基本英语”单词（或解释性词组）.

据统计，全球超过 55%的网页内容为英语[7]，而 45%的互联网用户年龄在 25 岁以下. 我们在 Facebook[8]上进行了面向日本、韩国、新加坡、印度、俄罗斯、斯诺文尼亚、巴西等国家高中生的调查. 结果显示，这些国家中，大多数都在中学阶段普及了英语教学，高中生可以理解简单的英语语句[9]. 在英语-本国语机器翻译难度最大的亚洲国家，英语语法是英语教学的中心，因此英语的句法对于亚洲学生并不特别困难. 相较而言，英语词汇对于非母语学习者和英语母语者都有很大难度[10].

从英语语言学的角度看，英语在历史演化过程中，丢失了印欧语系其他语言常见的语法特征[11]；此外，英语的语序较为固定. 这些因素导致英语语法较为简单. 但同时，英语不断吸收其他语言中的词汇，目前估计英语已有超过 100 万词[12]. 很多相同或相近的语义可以用语源不同的多个词语表示[13]. 综上，英语的难度主要体现在词汇方面.

鉴于此，我们提出了“英文文本机器简化”的方案. 该思路巧妙避免了复杂而不具有普遍性的句法差异处理过程.

将“基本英语”作为全球辅助语言的观念早在 20 世纪 30 年代即有人提出. 目前，“基本

[7] 见 http://w3techs.com/technologies/overview/content_language/all
[8] Facebook（http://www.facebook.com）是一个在全世界范围内流行的社交网站.
[9] 在有些欧美国家（如俄罗斯），大城市注重英语教育，而小城市或乡村缺乏英语教育. 但这些国家的语言在机器翻译处理时较为简便.
[10] 与《纽约时报》词汇难度相当的 SAT（美国大学入学考试）考试阅读部分，2012 年考生（多为美国高中生）的平均分只有 496 分（满分 800 分），考察词汇的问题得分普遍较低.
[11] 例如，名词的性、格特征，动词的屈折变化特征都较弱.
[12] 见 http://www.theguardian.com/books/2009/jun/10/english-million-word-milestone、http://www.languagemonitor.com/new-words/number-of-words-in-the-english-language-1008879/
[13] 如表示‘昂贵’这一语义的单词有 dear、expensive、costly、overpriced、exorbitant、extortionate 等.

英语”一项较为成功的应用是美国之音（Voice of America, VOA）的“特殊英语”（Special English）广播. 此外，如今英语的应用范围比其他任何一种语言都要广，英语不仅限于某个特定的民族、文化或者政体，这种多元性也使得英语易被众多不同母语的使用者所接受.

需要指出，该方案并非面向全球所有用户，而是专门为机器翻译目前无法较为准确地处理其母语（例如汉语和阿拉伯语）的互联网用户设计的，是一种可靠的过渡方案.

4 “基本英语”词汇数目的确定

4.1 简化方案的假设与单词的计数

为了定量分析“基本英语”中应涵盖的词汇数 n，我们假设非母语的互联网使用者对英语单词的熟悉程度与其词频正相关. 这里，我们将某单词的所有屈折词缀（inflectional affix）形式看做 1 个单词，但将单词加不同派生词缀（derivational affix）的形式看做多个单词. 此外，单词的多个义项按词类划分为多个不同单词.

4.2 语料库的选择

为了得到比较准确的词频信息，我们选择了收录 464020256 词的 Corpus of Contemporary American English（当代美国英语语料库，COCA）公布的词频信息（见附录）.

4.3 对词频与次序关系的分析

图 1 表示了 COCA 中最常见 5000 词的词频 f 与次序（按词频从高到低排列）r 的关系.

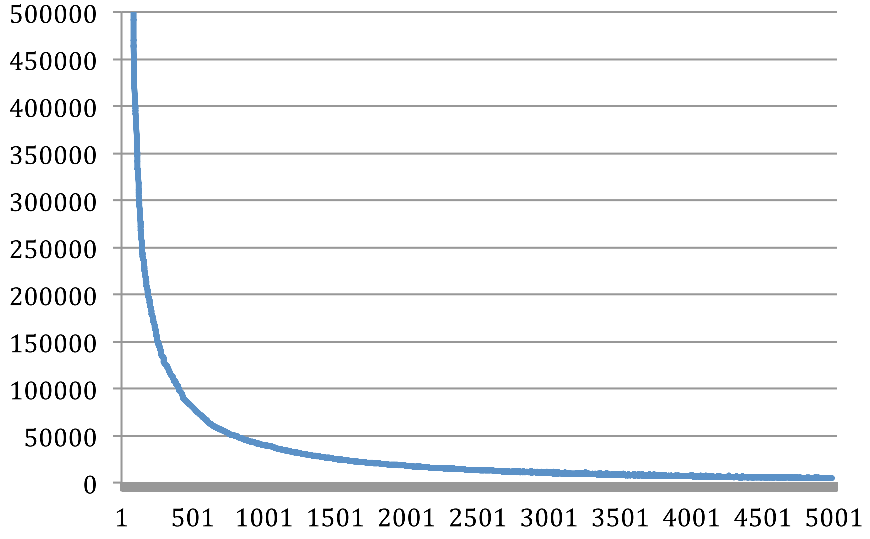


图 1 COCA 中最常见 5000 词的 f - r 图象

可以看出，f 与 r 间存在类似负指数函数型关系. 令

$$f = ar^{-b},$$

两边取对数（本文中 lg 指常用对数 $\log_{10}$），得

$$\lg f = \lg a - b \lg r.$$

作出 $\lg f$ - $\lg r$ 图象（图 2）. 可以看出，$\lg r > 1.4$ 时，线性拟合较好，而 $\lg r \le 1.4$ 时，误差较大. 因此，我们手动计算了次序前 25 的词语（$\lg 25 = 1.3979$），其词频总数

$$F(25) = 128431737,$$

覆盖率（占 COCA 总文本比例）

$$c(25) = \frac{F(25)}{464020256} = 27.6780\%.$$

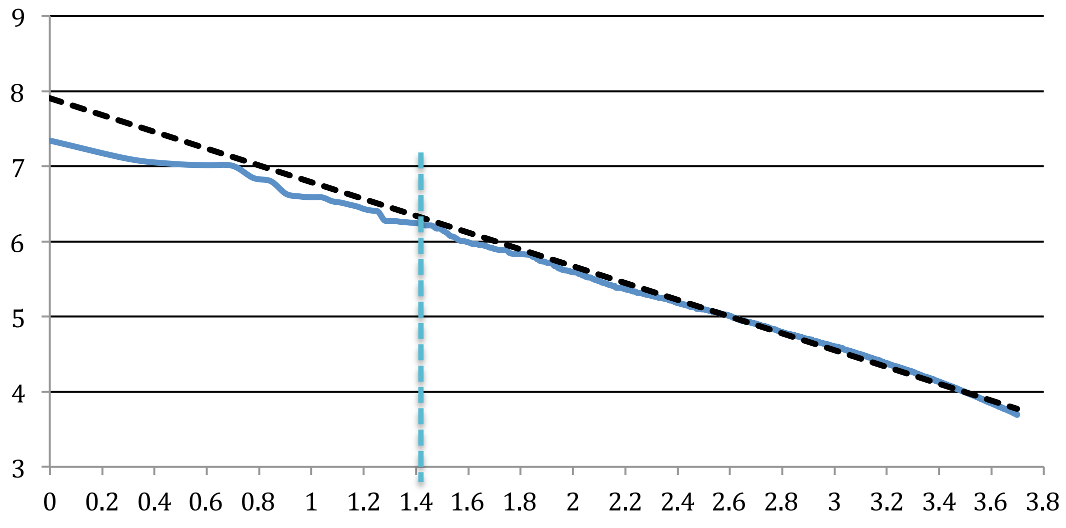


图 2 COCA 中最常见 5000 词的 $\lg f$ - $\lg r$ 图象及其线性拟合线

对 $\lg r > 1.4$ 的数据进行再次拟合（图 3），得出 $\lg f$ - $\lg r$ 的关系为

$$\lg f = 7.9555 - 1.1326 \lg r.$$

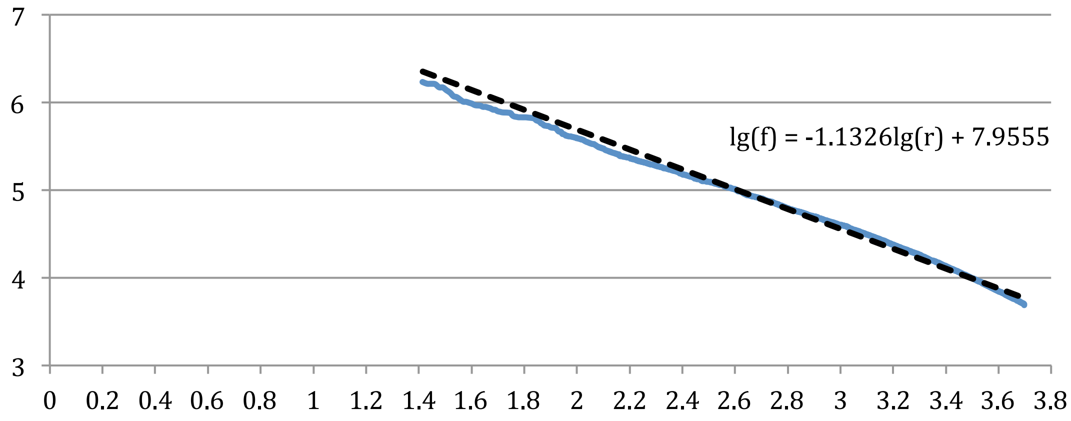


图 3 对 $\lg r > 1.4$ 的数据的再拟合

整理，得

$$f = \frac{9.0261 \times 10^7}{r^{1.1326}}, r \geq 26.$$

1949 年，语言学家齐夫提出，自然语言语料库中，某单词的词频与其在词频表中的次序成反比（齐夫定律，Zipf’s Law），即

$$f \propto \frac{1}{r}.$$

可见，我们得出的 f - r 经验公式与齐夫定律大致吻合.

4.4 对“基本英语”词汇数目与文本覆盖率关系的分析

由上节内容可得，COCA 中次序排名前 n 的单词词频总数

$$F(N)=F(25)+\sum_{i=26}^{n}\frac{9.0261\times10^{7}}{i^{1.1326}},n\geq26,$$

又 $F(25)=128431737$，$c(25)=27.6780\%$，因此，COCA 中次序排名前 n 的单词的覆盖率

$$c(n)=\frac{F(n)}{464020256}=c(25)+\frac{9.0261\times10^{7}}{4.6402\times10^{8}}\sum_{i=26}^{n}\frac{1}{i^{1.1326}}$$
$$=27.6780\%+1.94520\times10^{-1}\sum_{i=26}^{n}\frac{1}{i^{1.1326}},n\geq26.$$

为了判断 n 的合适取值，我们需研究函数 $c(n)$ 的性质，尤其是 $\sum_{i=26}^{n}\frac{1}{i^{1.1326}}$ 的性质. 令

$$A_n=\sum_{i=1}^{n}\frac{1}{i^{p}},\quad p=1.1326.$$

考虑用积分来近似，令

$$B_n=\int_1^n\frac{1}{x^p}dx=\frac{1-n^{1-p}}{p-1},\quad R_n=A_n-B_n.$$

由于函数 $\frac{1}{x^p}$ 在 $(0,+\infty)$ 上单调减，所以

$$B_n\leq\sum_{i=1}^{n-1}\frac{1}{i^p}<A_n,$$

即

$$R_n>0.$$

另一方面，

$$R_{n+1}-R_n=\frac{1}{(n+1)^p}-\int_n^{n+1}\frac{1}{x^p}dx=\int_n^{n+1}\left(\frac{1}{(n+1)^p}-\frac{1}{x^p}\right)dx.$$

由函数 $\frac{1}{x^p}$ 在 $(0,+\infty)$ 上单调减可知，对于 $n \le x \le n+1$，有

$$\frac{1}{(n+1)^p}-\frac{1}{x^p}\le 0.$$

因此

$$R_{n+1}-R_n=\int_n^{n+1}\left(\frac{1}{(n+1)^p}-\frac{1}{x^p}\right)dx<0,$$

即数列 R_n 单调减. 又由于 $R_n>0$，所以由单调数列定理可知数列 R_n 收敛.

使用计算机估算得出 $\lim_{n\to+\infty} R_n \approx 0.587$. 因此，当 n 较大时，

$$A_n \approx B_n+0.587=\frac{1-n^{1-p}}{p-1}+0.587=-\frac{n^{-0.1326}}{0.1326}+8.128.$$

综上可得，当 n 较大时，

$$\begin{aligned}
&c(n)\\
&=27.6780\%+1.94520\times10^{-1}\sum_{i=26}^{n}\frac{1}{i^{1.1326}}\\
&\approx 27.6780\%+1.94520\times10^{-1}\left(-\frac{n^{-0.1326}}{0.1326}+8.128-\sum_{i=1}^{25}\frac{1}{i^{1.1326}}\right)\\
&=-\frac{1.46697}{n^{0.1326}}+1.23152.
\end{aligned}$$

4.5 “基本英语”词汇数目的确定

我们在确定“基本英语”词汇数目 n 时有两方面的考量：

1、这 n 个词汇应能够覆盖绝大多数的正常英语文本. 如此，次序大于 n 的单词比例较低，且需要简化的内容可以较为顺畅地用次序小于 n 的单词表达.

令 $c(n)\ge 70\%$，带入表达式，即有 $n\ge 2114$. 又由图 3 可知，上节中 $c(n)$ 的近似公式结果偏大，因此可估计 $n\in[2000,2500]$ 时，大致有 $c(n)\in[66\%,70\%]$. 这个范围是较为合适的.

2、词汇数目大于 n 时，覆盖率不会显著增加，但词汇数目小于 n 时，覆盖率会显著

减少.

分别作出 $c(n)$ 和其导数 $\frac{dc(n)}{dn}$ 的图象

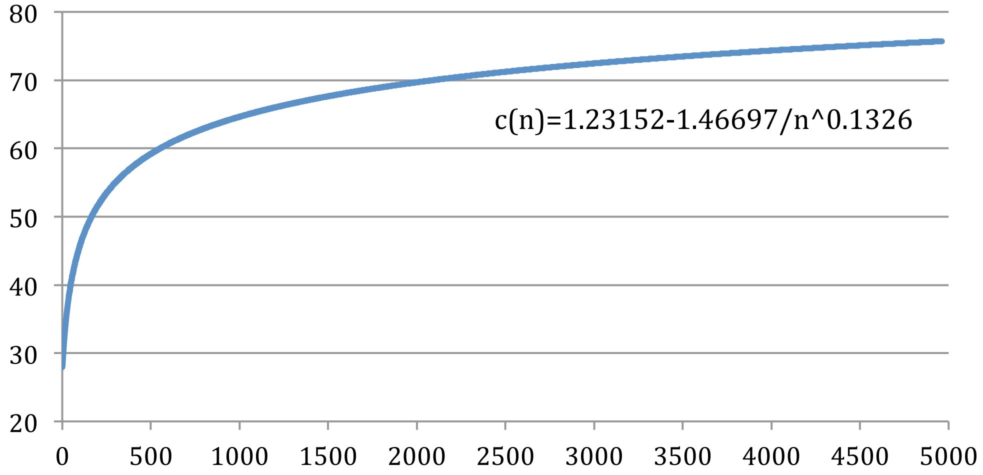


图 4 $c(n)$ (百分比)- n 图象

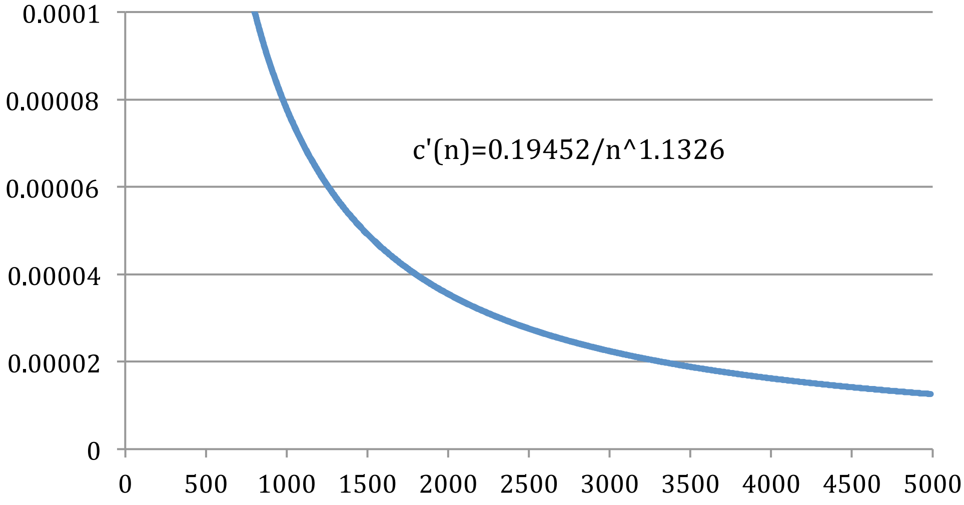


图 5 $\frac{dc(n)}{dn}$ - n 图象

从图 5 中可以看出，在区间 $[2000,3000]$， $c(n)$ 的增长速率明显减缓，这一点在图 4 中也得到了验证.

综上，我们将设计 3 个“基本英语”的词汇数目选项，2000 词、2500 词和 3000 词. 用户将可以自行选择适合自己的标准.

4.5 其他参考标准

除了进行语料库词频统计以确定“基本英语”词汇数目，我们还参考了英语作为第二语言（ESL）教学中常用的 3 本英语学习者词典[14]，三者分别使用 2000—3000 个基础英文词汇（这些词汇被认为是学习者已经掌握的）为陌生词汇释义，这与“英文文本及其简化”方案的基本思想是相通的. 这些词典在全球广受英语学习者好评的事实从一个侧面反映出我们选取 $n \in [2000,3000]$ 的方案具有可行性.

4.6 小结

要实现英文文本机器简化，首先需确定“基本英语”的词汇数目. 我们通过检验 COCA 词频数据，将“基本英语”词汇数目确定在 2000 至 3000 个，这与 ESL 教育中常见的基础词汇范围相符.

5 原始文本中非“基本英语”词汇的替换方法

5.1 确定“基本英语”对原始文本词汇的表述

5.1.1 简化（释义）的原则

对于非“基本英语”词汇（次序大于 n 的词汇），大多数需要用“基本英语”词汇进行简化. 简化时，理想情况下须依次遵循以下原则. 这三项原则不能同时满足时，应按照 1、2、3 的顺序予以取舍.

1、可替换性——“基本英语”的简化结果须可直接替换原文，而不造成语法或语义上的损失.

[14] 分别是《牛津高阶英语词典》（Oxford Advanced Learner's Dictionary, 8th edition）、《朗文当代高级英语辞典》（Longman Dictionary of Contemporary English, 5th edition）和《韦氏高阶美语词典》（Merriam-Webster’s Advanced Learner’s Dictionary, 1st edition）.

2、非歧义性——“基本英语”的简化结果语义必须明确.

3、简洁性——“基本英语”的简化结果须简单明了，尽可能使用一个单词简化另一个单词.

原则 1 和原则 3 在词典编纂的实践中很难实现，因为许多语法性词语（例如介词）无法用替换法定义. 但由于这里需要简化的都是次序大于2000的词语,因此在实际简化过程中，很多词语都是可通过直接替换实现.

5.1.2 简化（释义）的方法和范例

英语 100 万单词中，受过良好教育的大学毕业生大约认识 75000 词. 因此，我们的简化可以局限在次序 $2000 \le n \le 10000$ 的单词中. 首先，我们需要手动构建一部分正常英语-“基本英语”词典，在此过程中应参考英语母语者和 ESL 使用的词典，以确保最大程度的准确性. 以下是部分词条内容：

正常英语	词类	“基本英语”2000	2500	3000
aardvark	n	aardvark (large African animal)		
abandon	v	give up		
	v	leave		
	n	unlimited freedom		
abase	v	put down	lower	
aback	i	by surprise		
	i	toward the back		

需要注意，并非所有简化替代的内容皆须取自“基本英语”的词汇范畴，对此将在 5.3 节说明. 按 1 人 1 天可以编写 40 个词条计算,5 人可以在 50 天内完成 10000 条词条编写. 包括校验和小范围测试在内，2 至 3 个月可以完成初期的词典编写任务. 从词典中可以看出，当选用 3000 词或 2500 词模式时，2500 词释义优先级高于 2000 词释义. 此外，句法中心词（syntactic head）被特殊标记（这里用下划线表示），这有助于 5.3 节对屈折词缀的处理.

5.1.3 维基式公测

根据当次序 $2000 \le n \le 12000$ 词汇的内建词典已经全部完成时，根据 4.4 节求出的 $c(n)$ 表达式，覆盖率已达 80%. 此时机器简化程序可以开始进行公共测试. 对于无法精准简化的词语，公共测试期间可以以相同几率呈现多个简化版本. 用户可以通过投票方式表达对简化的满意度，满意度较高的翻译将被给予较大的权重. 用户还可以对未能简化的单词进行自行简化. 算法会吸收用户的建议，众多用户提交的简化方案将被算法采纳. 这样，编写后续词典的进程将大大加快，简化的质量也将逐步提高.

5.2 确定语义合理性的统计学模型及其评估

5.2.1 n-gram 语言模型

本文的第 2 部分提出，机器翻译中的词汇差异处理可以采用基于统计学模型的算法解决. IBM 1 模型无法联系上下文分析语义. 因此，我们采用 n-gram 模型分析英文文本机器简化算法中涉及到的语义识别问题.

文本中连续出现的某 n 个单词称为一个 n-gram. 例如在句子“That's one small step for a man, a giant leap for mankind.”中，small step for 是一个 3-gram，step for a man 是一个 4-gram，句中的任一单词都单独构成一个 1-gram.

n-gram 语言模型基于这样一种假设：每个词出现的概率只与在它之前的至多 n-1 个词有关，与其它任何词都不相关. 在这样的假设下，我们可以从大量语料中统计得到每个 n-gram 出现的频率，用来计算单词出现的概率，并进行进一步的文本简化处理.

n-gram 语言模型的优点在于，训练模型时只需要在大量语料中统计各个 n-gram 出现的次数，过程相对简单. 此外，在资源充足的情况下，只需要增加参数 n 的大小，就能使得模型的准确度有所提升.

在实际应用中，n-gram 语言模型被广泛运用于计算语言学，例如统计自然语言处理.

5.2.2 n-gram 模型的问题

n-gram 语言模型最大的问题在于可能的 n-gram 数量非常多. 假设词典大小仅为 V=10000，并取 n=3，可能的 n-gram 数量将会达到 $V^n=10^{12}$.

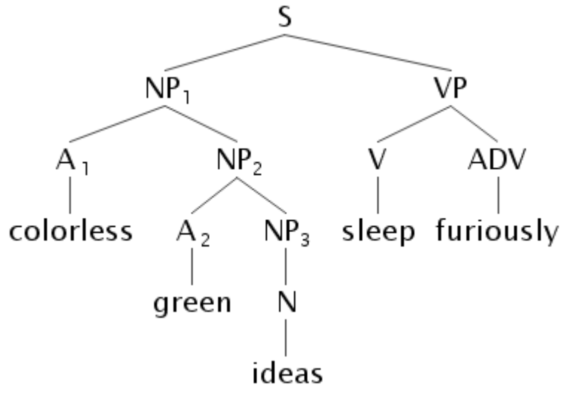
在实际应用中，计算机不仅没有足够的内存来存储这么多数据，而且运算速度也不足以快速处理这些数据. 更重要的是，为训练 n-gram 语言模型，需要对规模比 V^n 更大的语料库进行统计，这也是无法实现的.

5.2.3 n-gram 模型的改进

在自然语言中 n-gram 是非常稀疏的. 如果统计得出 V^n 个 n-gram 出现的频率，则可以肯定其中大部分频率都等于 0.

我们考虑随机选择 n 个英语单词排列在一起. 首先，得到的词组可能是不符合语法规则的，例如*VI N（*不及物动词 名词）[15]词组，或者在某个不能用作定语的形容词之后加上了名词.

此外，即使符合语法规则，词组也很可能没有实际语义. 乔姆斯基在《句法结构》中举过“Colorless green ideas sleep furiously”这个例子. 该句完全符合英语的语法规则：



[15] 符号*表示不符合语法.

但是在这个句子中，不仅 colorless 和 green 矛盾，而且使用形容词 colorless 或 green 形容 ideas，以及用副词 furiously 修饰 sleep 都是罕见的，所以这个句子几乎不可能出现在正常的文本中.

因此，如果我们利用 n-gram 的稀疏性，只统计并存储最经常出现的一部分 n-gram 语法，这样的语法的数量是可以接受的.

对于不常出现的 n-gram，我们使用一种“回退”的方式计算概率. 以 5.2.1 节的语句为例，我们要计算 4-gram step for a man 出现的概率，记为 P(step for a man). 但是 step for a man 这个 4-gram 可能在语法中并没有记录. 这时我们去掉第 1 个单词，得到 for a man，这个 3-gram 是有记录的，因此 P(for a man)等于该 3-gram 出现的频率，记为 F(for a man).

但是，因为我们去掉了第 1 个单词“step”，才使得 P(step for a man)增加为 F(for a man)，所以还需要乘一个“回退权”来表示去掉 step 的损失，记为 B(step for a). 即 P(step for a man)=F(for a man)·B(step for a). 对于每个 n-gram，回退权 B 是一个和频率 F 一同记录的参数. 如果某个 n-gram 没有被记录，那么我们认为它的回退权为 1.

如果去除第 1 个单词之后的 n-gram 仍然没有记录，可以继续去掉剩下的第 1 个单词，并乘上对应的回退权，直到得到一个在语法中有记录的 n-gram 为止. 以 5-gram“colorless green ideas sleep furiously”为例：

P(colorless green ideas sleep furiously)=P(green ideas sleep furiously)·B(colorless green ideas sleep)

P(green ideas sleep furiously)=P(ideas sleep furiously)·B(green ideas sleep)

P(ideas sleep furiously)=P(sleep furiously)·B(ideas sleep)

P(sleep furiously)=P(furiously)·B(sleep)

P(furiously)=F(furiously)

综合以上 5 个等式得出：

P(colorless green ideas sleep furiously)=B(colorless green ideas sleep)·B(green ideas sleep)·B(ideas sleep)·B(sleep)·F(furiously).

在上述例子中，为了计算 P(colorless green ideas sleep furiously)，我们连续回退了 4 次得到 P(furiously)，而 1-gram 的概率 P(furiously)就是该单词出现的频率，即 F(furiously).

5.3 n-gram 语言模型的应用

我们可在 n-gram 语言模型的基础上加以修改，以处理英文文本机器简化中的语义识别问题.

首先要解决的问题是参数 n 的选择. 网络上收集到的 n-gram 公开统计资料中，最大的 n 为 5. 另一方面，我们相信，一个单词的前后 4 个单词足以得出它的出现概率，因此下面的讨论中取 n=5.

设我们在分析一个包含 L 个单词的句子 $W_1,W_2,W_3,...,W_L$ 中的第 i 个词 W_i，我们不妨假设 $5\le i\le L-4$，这样 W_i 前后都至少有 4 个词. 考虑 W_i 在“基本英语”中的 m 种可能的替换 $V_1,V_2,...,V_m$. 为辨别出最合适的替换，我们给每一个替换 V 一个评分 Score(V)，求出 $Score(V_1),Score(V_2),...,Score(V_m)$之后，选择其中得分最高的作为输入文本中单词 W_i 的最佳替换. 下面我们重点讨论评分函数 Score(V)的设计.

这个评分的目的是反映替换后语义的合理性. 因此，我们将 Score(V)定义为：把 W_i 替换为 V 后，整个句子出现的概率，记为 $P'(W_1,W_2,...,W_{i-1},V,W_{i+1},...,W_L)$.

由于 n-gram 模型的假设，我们只需要考虑 V 前后的 4 个单词，

$$P'(W_{i-4},W_{i-3},W_{i-2},W_{i-1},V,W_{i+1},W_{i+2},W_{i+3},W_{i+4}).$$

因为对于当前的分析而言，参数中只有 V 是变量，所以我们只需要考虑与 V 相关的 n-gram 出现的概率. 根据乘法原理，这个概率为

$$P'(W_{i-4},W_{i-3},W_{i-2},W_{i-1},V)\cdot P'(W_{i-3},W_{i-2},W_{i-1},V,W_{i+1})\cdot\ldots\cdot P'(V,W_{i+1},W_{i+2},W_{i+3},W_{i+4}).$$

之前提到，在计算 $P(X_1,X_2,...,X_5)$时，如果 5-gram X_1,X2,...,X_5 不存在，一般采用“回退”的方式计算. 这个方法涉及到一个参数“回退权”. 由于公开的统计资料有限，我们使用另一种处理方式，这个方法只需要使用每个常用 n-gram 的频率 F.

回想使用“回退权”的目的，是表示从 n-gram 中去掉若干个词带来的损失. 我们不妨反过来考虑，对“没有去掉词”的 n-gram 进行“奖励加分”. 我们规定

$$P'(X_1,X_2,X_3,X_4,X_5)=F(X_1,X_2,X_3,X_4,X_5)\cdot F(X_2,X_3,X_4,X_5)\cdot F(X_3,X_4,X_5)\cdot F(X_4,X_5)\cdot F(X_5)$$

在上式中，如果某个 n-gram 并没有在记录中出现，那么对应的频率记为常数 ε，ε 应当远小于最小的频率并且大于 0.

我们考虑计算两个 5-gram “the owner of the shop”与“colorless green ideas sleep furiously”出现的概率 P’. 根据上述公式：

P’(the owner of the shop)=F(the owner of the shop)·F(owner of the shop)·F(of the shop)·F(the shop)·F(shop)

P’(colorless green ideas sleep furiously)=F(colorless green ideas sleep furiously)·F(green ideas sleep furiously)·F(ideas sleep furiously)·F(sleep furiously)·F(furiously)

第一式中的 5-gram the owner of the shop 比较合理，因此等式右侧的五个 n-gram 都在语法中有记录.

在第二式中，因为涉及的 2-gram,3-gram,4-gram 和 5-gram 非常罕见，在语法资料中都没有记录，所以依照我们的规则得到

$$P'(\text{colorless green ideas sleep furiously})=\varepsilon^4\cdot F(\text{furiously})$$

由于 ε 是一个很小的常数，所以得到的结果是 P’(colorless green ideas sleep furiously)远小于 P’(the owner of the shop)，是合理的.

5.4 例外及其解决办法

在简化非“基本英语”单词时，必须注意一些例外情况.

5.4.1 屈折词缀

语料库的词频统计并不包括含有屈折词缀形式. 因此，算法在离析词汇时，会一并检测出现的屈折词缀词汇单位，判断该词缀对应的屈折变化，并要求对应简化的句法中心词（syntactic head）进行相同的屈折变化. 这个过程由算法中的一系列英语形态音系学上下文无关规则实现.

5.4.2 专有名词和术语

专有名词和普通名词一样被计入语料库统计. 因此，在内建词典中，专有名词永远被简化为其本身. 术语通常由普通名词组成，对于 1 个单词构成的术语，词典中应该具有该单词作为术语的义项. 特别地，非英语的术语由于不会被计入语料库中，所以在输出时会保留其原形.

5.4.3 其它较为专业的名词

较为专业的名词通常在简化解释中保留其原型，同时在简化（解释）中加以简单描述，例如 5.1.2 节词典中对 aardvark 一词的简化. 经过这样的简化，一般用户可以得知这个名词所指的大致内容，而需要专业知识的用户也可以继续查询这方面的信息.

5.4.4 成语动词

成语动词（phrasal verb）是由一个（通常较为普遍的）动词和介词/副词连用的动词形式. 由于这样的动词和介词/副词都较为常见，理论上不会在简化时发生变化. 但是成语动词

的语义对于英语非母语者可能是陌生的，因此简化（解释）中应避免不常见的成语动词，但这并不意味着不能使用常见的成语动词（例如 5.1.2 节词典中的 give up）.

5.4.5 对简化（解释）可用词汇的附加说明

简化所用的词汇原则上应来源于相应数量的“基本英语”词汇. 但大多数情况下，由多个“基本英语”词汇构成的、语义为两语素简单叠加的合成词也可被接受.

5.5 小结

英文文本机器简化算法内建词典的简化解释应该遵循可替换性、非歧义性和简洁性的原则. 首批 10000 个单词简化完毕后进行程序公测，并通过维基式的学习不断完善词典和算法. 我们采用修正后的 n-gram 语言模型，对常见的 n-gram 语法加以储存，对不常见的 n-gram 语法采用“回退”方式处理，以解决词汇语义差异和义项选择的问题. 屈折词缀和其他例外应综合运用语言学规则和词典简化（释义）方法处理.

6 算法（伪代码）

6.1 函数说明

本算法涉及 3 个函数.

函数 Part_of_Speech_Tagging 作用为“词类标记”，在 6.2 节（“基本英语”词汇判断算法）中通过与语料库语料对比分析实现.

函数 Inflectional_Affix 作用为“识别（并去除）屈折词缀”; 函数 Apply_Inflectional_Auffix 作用为“在句法中心词上进行屈折变化”，这 2 个函数需要通过英语的形态音系学规则实现. 例如，对名词复数后缀的形式判断依次基于如下规则：

1、{-z}→|-s|　/ 前置辅音=清音 AND 非嘶音

2、{-z}→|-ɨz| / 前置辅音=嘶音

3、{-z} → |-z| / 其它

第1条和第3条规则输出-s，而第2条规则输出-es. 这样的有序规则容易在算法中实现.

6.2 “基本英语”词汇判断算法

```
#Word_Paraphrase(W , Pos)返回单词 W 在词类 PoS 下的释义列表
Word_Paraphrase (W , Pos)
    L ← 空列表
    n ← 单词 W 在词类 PoS 下的义项总数
    for i = 1 to n do
        if 第 i 个义项存在“3000 词释义” then
            在 L 末尾添加第 i 个义项的“3000 词释义”
        else if 第 i 个义项存在“2500 词释义” then
            在 L 末尾添加第 i 个义项的“2500 词释义”
        else
            在 L 末尾添加第 i 个义项的“2000 词释义”
        endif
    返回 L
```

6.3 n-gram 语言模型算法

```
#Sentence_Score(S)返回句子 S 的语义合理性评分，句中单词依次记为 W1,W2,…
Sentence_Score(S)
    n ← S 包含的单词数
    Score ← 1
    for i = 1 to n do
        #start 表示以第 i 个词为结尾的 5-gram 的起始位置，如果 i<4 则 start 为 1
        if i < 4 then
            start ← 1
        else
            start ← i - 4
        endif
        for j = start to i do
            if n-gram “Wj, Wj+1, …, Wi” 在语法规则中存在 then
                #得分乘上出现的频率
                Score ← Score * F(Wj , Wj+1 , … , Wi)
            else
                #得分乘上常数 ε
                Score ← Score * ε
            endif
    返回 Score
# Sentence_Simplification(S)返回句子 S 简化后的结果，句中单词依次记为 W1,W2,…
```

Sentence_Simplification(S)

```
    记录 S 中标点符号并从 S 中去除所有标点
    n ← S 包含的单词数
    #  函数 Part_of_Speech_Tagging 标记 S 中各个单词的词类，记为 PoS_1,PoS_2,…, PoS_n
    Part_of_Speech_Tagging(S)
    for i = 1 to n do
        #函数 Inflectional_Affix 返回一个二元组，分别为单词的屈折词缀和原型
        (IS_i , W_i) ← Inflectional_Affix(W_i)
    S’ ←  空文本
    for i = 1 to n do
        #  查找单词 W_i 在词类 PoS_i 下的所有释义
        L ← Word_Paraphrase(W_i , PoS_i)
        Len ←  列表 L 的长度
        MaxScore ← 0
        # Best 记录当前的最优释义，初始值为空
        Best ← NULL
        for j = 1 to Len do
            tmp ←  将 S 中 W_i 替换为释义 L_j 之后的句子
            #如果替换后的得分更高，更新最优释义
            if Sentence_Score(tmp) > MaxScore then
                MaxScore ← Sentence_Score(tmp)
                Best ← L_j
        #  函数 Apply_Inflectional_Affix 返回：释义 Best 的句法中心词添加屈折词缀
IS_i 后的结果
        在 S’的末尾添加 Apply_Inflectional_Affix(Best , IS_i)
    在 S’中添加上原句的标点符号
    返回 S’
```

6.4 词汇简化算法

Text_Simplification(T)为文本简化算法的主函数，返回输入的英文文本 T 的简化结果

Text_Simplification(T)

```
    将文本 T 划分为 m 个句子 S_1,S_2,...,S_m
    T’ ←  空文本
    for i = 1 to m do
        #  将 T 逐句简化后合并
        在 T’的末尾添加 Sentence_Simplification(S_i)
    返回 T’
```

6.5 小结

本算法分为 3 个步骤：判断词汇是否需要简化、考察需简化词汇的合理义项、实行词汇

简化. 在算法和词典的基础上，英文文本机器简化的方案将得以最终实现.

7 参考资料

Chomsky, N. 1969. *Syntactic structure*. The Hague.

Corpus.byu.edu. 1990. *Corpus of Contemporary American English (COCA)*. [online] Available at: http://corpus.byu.edu/coca/ [Accessed: 30 Aug 2013].

Crystal, D. 1997. *English as a global language*. Cambridge [England]: Cambridge University Press.

Crystal, D. 1995. *The Cambridge encyclopedia of the English language*. Cambridge [England]: Cambridge University Press.

Hutchins, J. 2006. *The first public demonstration of machine translation: the Georgetown-IBM system, 7th January 1954*. [e-book] Available through: Google hutchinsweb.me.uk/AMTA-2004-exp-rv.pdf [Accessed: 2 Aug 2013].

Koehn, P. 2010. *Statistical machine translation*. Cambridge: Cambridge University Press.

Landau, S. 1984. *Dictionaries*. New York: Scribner.

Ngrams.info. 2013. *N-grams: based on 450 million word COCA corpus*. [online] Available at: http://www.ngrams.info/ [Accessed: 30 Aug 2013].

Payne, T. 1997. *Describing Morphosyntax A Guide for Field Linguists*. Cambridge: Cambridge University Press.

Research.microsoft.com. 2013. *Microsoft Web N-gram Services - Microsoft Research*. [online] Available at: http://research.microsoft.com/en-us/collaboration/focus/cs/web-ngram.aspx [Accessed: 30 Aug 2013].

Wordfrequency.info. 2013. *Word frequency: based on 450 million word COCA corpus*. [online] Available at: http://www.wordfrequency.info/ [Accessed: 30 Aug 2013].

附录

Corpus of Contemporary American English（COCA）最常见 500 词表

（基于 http://www.wordfrequency.info/的 5000 词表）

次序(r)	单词	词类	词频(f)
1	the	a	22038615
2	be	v	12545825
3	and	c	10741073
4	of	i	10343885
5	a	a	10144200
6	in	i	6996437
7	to	t	6332195
8	have	v	4303955
9	I	p	3978265
10	it	p	3872477
11	to	i	3856916
12	that	c	3430996
13	for	i	3281454
14	you	p	3081151
15	he	p	2909254
16	with	i	2683014
17	do	v	2573587
18	on	i	2485306
19	say	v	1915138
20	this	d	1885366
21	they	p	1865580
22	we	p	1820935
23	his	a	1801708
24	but	c	1776767
25	at	i	1767638
26	that	d	1712406
27	not	x	1638830
28	from	i	1635914
29	n't	x	1619007
30	by	i	1490548
31	she	p	1484869
32	or	c	1379320
33	as	c	1296879
34	what	d	1181023
35	go	v	1151045
36	their	a	1083029
37	can	v	1022775
38	who	p	1018283
39	get	v	992596
40	her	a	969591
41	if	c	933542
42	would	v	925515
43	my	a	919821
44	know	v	892535
45	all	d	892102
46	about	i	874406
47	make	v	857168
48	as	i	829018
49	will	v	824568
50	up	r	795534
51	there	e	784528
52	think	v	772787
53	year	n	769254
54	one	m	768232
55	time	n	764657
56	so	r	756550
57	me	p	709623
58	people	n	691468
59	which	d	685982
60	when	c	678626
61	out	r	678603
62	them	p	677870
63	just	r	677711
64	him	p	677707
65	some	d	674193
66	take	v	670745
67	into	i	668172
68	see	v	663645
69	your	a	659622
70	come	v	628254
71	could	v	617932
72	now	r	605997
73	than	c	579757
74	like	i	568850
75	other	j	547799

76	then	r	543977	120	after	i	311902
77	its	a	539719	121	should	v	310265
78	how	r	538893	122	call	v	308050
79	our	a	525107	123	school	n	304183
80	more	r	517536	124	world	n	303506
81	want	v	514972	125	over	i	300349
82	these	d	513864	126	still	r	296953
83	two	m	511027	127	try	v	294023
84	look	v	491707	128	last	m	289843
85	way	n	470401	129	in	r	285035
86	also	r	464606	130	ask	v	284632
87	first	m	463566	131	as	r	281483
88	because	c	438539	132	too	r	280396
89	new	j	435993	133	need	v	276744
90	day	n	432773	134	feel	v	275214
91	use	v	420781	135	state	n	272193
92	more	d	420170	136	when	r	268219
93	here	r	412315	137	three	m	266744
94	well	r	411776	138	between	i	264158
95	man	n	409760	139	really	r	263414
96	no	a	402222	140	never	r	262584
97	thing	n	400724	141	become	v	259102
98	her	p	397950	142	high	j	255936
99	find	v	395203	143	student	n	255047
100	very	r	391821	144	something	p	254910
101	tell	v	388155	145	most	r	246360
102	many	d	385348	146	much	d	244507
103	give	v	384503	147	family	n	243267
104	only	r	379574	148	out	i	242443
105	those	d	378007	149	mean	v	242198
106	one	p	369553	150	another	d	240646
107	back	r	367844	151	leave	v	240482
108	even	r	361067	152	own	d	240452
109	good	j	353973	153	let	v	240300
110	us	p	351088	154	put	v	237480
111	any	d	348100	155	on	r	236980
112	woman	n	341422	156	old	j	236577
113	through	i	340921	157	why	r	235442
114	child	n	333849	158	while	c	234555
115	there	r	333433	159	keep	v	231760
116	life	n	333085	160	group	n	229435
117	down	r	329409	161	talk	v	229429
118	may	v	324569	162	big	j	227169
119	work	v	318210	163	hand	n	225247

164	great	j	225005	208	today	r	183724
165	country	n	223138	209	happen	v	182714
166	same	d	222836	210	like	v	182341
167	turn	v	221392	211	always	r	179474
168	seem	v	219627	212	move	v	179388
169	begin	v	218617	213	believe	v	178397
170	problem	n	217728	214	point	n	177481
171	help	v	216082	215	hold	v	177368
172	American	j	214968	216	all	r	177317
173	start	v	213952	217	million	m	176895
174	where	c	213744	218	next	m	176306
175	every	a	212739	219	live	v	176144
176	might	v	209059	220	large	j	175611
177	about	r	208550	221	bring	v	174366
178	over	r	208260	222	study	n	174069
179	show	v	208037	223	before	i	172769
180	part	n	207861	224	room	n	172472
181	such	d	207065	225	without	i	172448
182	again	r	206895	226	must	v	171043
183	right	r	205250	227	home	n	170527
184	against	i	204379	228	lot	n	169570
185	company	n	203345	229	mother	n	169407
186	place	n	202427	230	eye	n	169150
187	case	n	200773	231	water	n	167666
188	system	n	200175	232	national	j	166359
189	week	n	199268	233	area	n	165812
190	few	d	197266	234	money	n	164794
191	most	d	197086	235	under	i	164766
192	each	d	196522	236	fact	n	164401
193	hear	v	196070	237	story	n	163582
194	program	n	195985	238	right	n	163259
195	where	r	194427	239	month	n	162685
196	question	n	192070	240	different	j	162411
197	so	c	191893	241	write	v	161824
198	government	n	191314	242	head	n	160131
199	during	i	190729	243	young	j	160011
200	Mr	n	188555	244	yes	u	157364
201	play	v	188328	245	issue	n	156417
202	work	n	187533	246	kind	n	155032
203	run	v	187325	247	job	n	154743
204	number	n	186005	248	business	n	154468
205	small	j	185463	249	book	n	154013
206	night	n	184511	250	word	n	152891
207	off	r	183854	251	side	n	152559

252	though	c	152182	296	meet	v	128737
253	provide	v	150879	297	almost	r	127907
254	black	j	150718	298	set	v	127369
255	four	m	150646	299	information	n	127331
256	little	j	149658	300	face	n	127291
257	house	n	149251	301	name	n	127139
258	long	j	149050	302	white	j	126760
259	far	r	148621	303	nothing	p	126717
260	sit	v	147185	304	minute	n	126660
261	both	d	146338	305	later	r	126495
262	game	n	146311	306	kid	n	126428
263	service	n	146122	307	right	j	126278
264	father	n	145051	308	once	r	126203
265	away	r	144713	309	continue	v	126029
266	political	j	144437	310	much	r	126029
267	important	j	144194	311	five	m	125571
268	around	i	143766	312	ago	r	125252
269	friend	n	142697	313	body	n	125165
270	after	c	142289	314	back	n	125006
271	however	r	142282	315	door	n	124993
272	long	r	142007	316	watch	v	124976
273	power	n	141357	317	best	j	124850
274	since	c	141264	318	learn	v	124346
275	stand	v	140937	319	real	j	124187
276	until	c	140819	320	several	d	124039
277	often	r	140731	321	least	r	123961
278	hour	n	138955	322	change	v	123183
279	among	i	138192	323	around	r	122789
280	line	n	135986	324	lead	v	122691
281	ever	r	135774	325	idea	n	122140
282	yet	r	135484	326	whether	c	121921
283	bad	j	134910	327	level	n	121704
284	member	n	134731	328	stop	v	121481
285	president	n	134203	329	understand	v	121354
286	end	n	134104	330	anything	p	120292
287	lose	v	134102	331	public	j	119825
288	law	n	133706	332	parent	n	119610
289	car	n	133571	333	follow	v	119425
290	include	v	133563	334	create	v	119419
291	pay	v	133133	335	together	r	119186
292	community	n	133057	336	such	i	119125
293	social	j	132899	337	art	n	117851
294	city	n	132684	338	add	v	117842
295	team	n	131489	339	war	n	117804

340	health	n	117762
341	only	j	117700
342	speak	v	117358
343	result	n	116277
344	sure	j	116186
345	teacher	n	116100
346	others	n	115771
347	already	r	115220
348	history	n	114904
349	allow	v	114892
350	research	n	114802
351	office	n	114791
352	within	i	114599
353	spend	v	114569
354	read	v	114094
355	morning	n	114002
356	walk	v	113787
357	education	n	113731
358	person	n	113650
359	party	n	112962
360	change	n	112426
361	open	v	111857
362	win	v	111478
363	girl	n	110409
364	guy	n	110409
365	grow	v	110020
366	moment	n	109720
367	himself	p	109288
368	low	j	108990
369	maybe	r	108421
370	early	j	108171
371	force	n	108005
372	although	c	107925
373	food	n	107728
374	policy	n	107601
375	before	c	107448
376	boy	n	107447
377	process	n	107341
378	foot	n	107285
379	remember	v	106879
380	reason	n	106863
381	offer	v	106473
382	both	r	106361
383	toward	i	105984

384	air	n	105932
385	enough	r	105880
386	across	i	105559
387	actually	r	105155
388	off	i	104122
389	love	v	103681
390	including	i	103650
391	second	m	103621
392	oh	u	103613
393	everything	p	103591
394	age	n	103402
395	yeah	u	103389
396	able	j	103171
397	music	n	102657
398	wait	v	102463
399	consider	v	101987
400	human	j	101224
401	buy	v	101105
402	appear	v	100671
403	market	n	100435
404	probably	r	99754
405	serve	v	99660
406	die	v	98376
407	experience	n	98106
408	home	r	97937
409	nation	n	97212
410	college	n	97038
411	stay	v	96933
412	fall	v	96908
413	build	v	96651
414	interest	n	96620
415	send	v	96613
416	use	n	96564
417	course	r	96224
418	cut	v	96012
419	sense	n	95896
420	plan	n	95824
421	someone	p	95608
422	expect	v	95566
423	effect	n	95216
424	behind	i	95047
425	death	n	93222
426	local	j	92970
427	kill	v	92660

428	suggest	v	92643	465	along	i	84926
429	reach	v	92375	466	arm	n	84865
430	development	n	91995	467	sometimes	r	84845
431	class	n	91323	468	develop	v	84835
432	remain	v	91319	469	relationship	n	84549
433	six	m	90571	470	heart	n	84536
434	economic	j	90392	471	price	n	84443
435	control	n	90301	472	decide	v	84035
436	voice	n	89379	473	better	j	83895
437	require	v	89280	474	according	i	83773
438	former	d	88930	475	whole	j	83756
439	care	n	88862	476	season	n	83743
440	little	r	88697	477	strong	j	83677
441	role	n	88666	478	wife	n	83601
442	thank	v	88574	479	report	n	83174
443	report	v	88138	480	model	n	82973
444	else	r	87876	481	value	n	82942
445	sell	v	87865	482	less	r	82930
446	major	j	87487	483	difference	n	82911
447	light	n	87427	484	mind	n	82808
448	field	n	87243	485	decision	n	82429
449	pull	v	87243	486	free	j	82090
450	rate	n	87224	487	finally	r	81951
451	perhaps	r	87060	488	federal	j	81826
452	raise	v	87036	489	return	v	81812
453	show	n	86828	490	international	j	81610
454	hard	j	86817	491	hope	v	81385
455	effort	n	86473	492	player	n	81358
456	late	j	86421	493	view	n	81338
457	drug	n	86231	494	society	n	81192
458	pass	v	86184	495	road	n	80987
459	police	n	85880	496	son	n	80895
460	up	i	85759	497	explain	v	80797
461	leader	n	85438	498	tax	n	80713
462	themselves	p	85256	499	join	v	80609
463	military	j	85152	500	drive	v	80476
464	possible	j	85084				

词类标记

a：冠词/形容词性物主代词
c：连词
d：限定词
e：表存在的 there
i：介词
j：形容词
m：数词
n：名词
p：代词
r：副词
t：不定式标记 to
u：感叹词
v：动词
x：否定词 not 和 n’t

作者介绍

曹起曈，南京外国语学校 2014 届学生，在 2013 年国际语言学奥林匹克竞赛（International Linguistics Olympiad）中获得荣誉提名，这是中国在该竞赛中获得的最高奖项.

乔明达，南京外国语学校 2014 届学生，在 2013 年国际信息学奥林匹克竞赛（International Olympiad in Informatics）荣获金牌，成绩位列全球第 2 名.